

**Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
"Национальный исследовательский университет  
"Высшая школа экономики"**

Факультет экономики, менеджмента и бизнес-информатики  
Кафедра информационных технологий в бизнесе

**Рабочая программа дисциплины  
Алгоритмы и структуры данных**

для образовательной программы «Программная инженерия»  
направления подготовки 09.03.04 Программная инженерия  
уровень бакалавриат

Разработчик программы  
Морозенко В.В., к.ф.-м.н., доцент, v.morozenko@mail.ru

Одобрена на заседании кафедры информационных технологий в бизнесе  
«25» августа 2016 г.  
И.о. зав. кафедрой информационных технологий в бизнесе  
О.Л. Викентьева \_\_\_\_\_

Утверждена Академическим советом образовательной программы «Бизнес-информатика» направления подготовки 38.03.05 Бизнес-информатика, образовательной программы «Программная инженерия» направления подготовки 09.03.04 Программная инженерия, образовательной программы «Информационная аналитика в управлении предприятием» направления подготовки 38.04.05 Бизнес-информатика  
«30» августа 2016 г., № протокола 8.2.2.1-28-09/05

Академический руководитель образовательной программы «Программная инженерия»

А.О. Сухов \_\_\_\_\_

Пермь, 2016

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.*



## 1 Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления подготовки 09.03.04 Программная инженерия, изучающих дисциплину «Алгоритмы и структуры данных».

Программа разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ по направлению подготовки 09.03.04 Программная инженерия, утвержденным ученым советом НИУ ВШЭ, протокол от 30.01.2015 №1;
- Основной образовательной программой высшего образования «Программная инженерия» направления подготовки 09.03.04 Программная инженерия;
- Объединенным учебным планом университета по образовательной программе «Программная инженерия» направления подготовки 09.03.04 Программная инженерия, утвержденным в 2015 г.

## 2 Цели освоения дисциплины

Целями освоения дисциплины «Алгоритмы и структуры данных» являются:

1. Развитие профессионального кругозора и алгоритмического мышления студентов.
2. Выработка у студентов навыков решения задач, требующих разработку и формализацию алгоритмов и использования основных структур данных.
3. Обучение студентов важнейшим теоретическим положениям информатики.

Для достижения поставленных целей при изучении дисциплины решаются следующие задачи:

- познакомить студентов с основными классами алгоритмов, их математическими моделями;
- дать знания о существующих эффективных алгоритмах для решения наиболее известных задач комбинаторной оптимизации, об их сложности и требованиям к памяти;
- познакомить с классификацией оптимизационных задач и алгоритмов для их решения, особенностями задач комбинаторной оптимизации большой размерности;
- дать представление о методах анализа сложности алгоритмов и доказательства их корректности.

Курс призван повысить общую эрудицию студентов, дать им возможность ориентироваться в данной предметной области, подготовить к применению теоретических знаний при решении различных задач оптимизации, при изучении и разработке средств поддержки принятия решений. Студенты должны получить знания о существующих эффективных алгоритмах, используемых в теории расписаний, методах их разработки и анализа.

## 3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Знать о различных методах классификации существующих алгоритмов; наиболее известные алгоритмы для работы с различными структурами данных; точные и приближенные подходы к решению типовых задач, возникающих в области программной инженерии; особенности точных, приближенных, эвристических, переборных, «жадных» алгоритмов.
- Уметь анализировать существующие алгоритмы с точки зрения их эффективности и применимости для решения прикладных задач; разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии; оценивать сложность разработанных алгоритмов и обосновывать их корректность.



- Иметь навыки (приобрести опыт) применения известных и разработки собственных алгоритмов для решения практических задач с учетом требований к точности, времени работы алгоритма и вычислительным ресурсам; формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам.

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ФГОС/ НИУ	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции
Способен обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности	ПК-4	Демонстрирует умение обосновывать предлагаемые решения, разрабатывать алгоритмы и программы, реализующие их	Контактные занятия проводятся в форме, предполагающей активное участие студентов в работе, обсуждение проблем и анализ решений, предлагаемых студентами и преподавателем на лекциях и практических занятиях
Способен выполнить начальную оценку степени трудности, рисков, затрат и сформировать рабочий график	ПК-7	Четко формулирует задачи, анализирует условия и обоснованно выбирает методы решения, уверенно интерпретирует результаты	Выполнение заданий с постепенным наращиванием требований к сложности, используемым методам и средствам решения
Способен оценивать временную и емкостную сложность программного обеспечения	ПК-13	Знает и может использовать на практике математический аппарат, формальные средства, лежащие в основе различных методов разработки алгоритмов и программ	Выполнение практических заданий с использованием языков C# и C++, тестирование разработанных программ с использованием различных методов и средств отладка
Способен использовать различные технологии разработки программного обеспечения	ПК-16	Показывает умение чтения исходного кода алгоритмов, решающих отдельные подзадачи в рамках одной исходной задачи	Получение формальных оценок и сравнение их с результатами, полученными при практической реализации
Способен применять основные методы и инструменты разработки программного обеспечения	ПК-17	Уверенно использует способы формального описания алгоритмов с применением математического аппарата. Может построить оценки и доказать свойства алгоритмов и программ с использованием формальных методов	Получение формальных оценок и сравнение их с результатами, полученными при практической реализации

#### 4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части профессионального цикла дисциплин (Major).

Изучение данной дисциплины базируется на следующих дисциплинах:

- Программирование.



- Дискретная математика.

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:

- способен вести исследовательскую деятельность, включая анализ проблем, постановку целей и задач, выделение объекта и предмета исследования, выбор способа и методов исследования, а также оценку его качества
- способен работать в команде
- способен применять основные концепции, принципы, теории и факты, связанные с информатикой при решении научно-исследовательских задач;
- способен к формализации в своей предметной области с учетом ограничений используемых методов исследования
- способен моделировать, анализировать и использовать формальные методы конструирования программного обеспечения
- способен использовать различные технологии разработки программного обеспечения
- способен применять основные методы и инструменты разработки программного обеспечения

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- Распределенные вычисления.
- Современные языки и технологии программирования.
- Функциональное и логическое программирование.
- Web-программирование.
- Семантические информационные системы.
- Командный проект по программной инженерии.

## 5 Тематический план учебной дисциплины

№	Название раздела	Всего часов	Контактные часы		Самостоятельная работа
			Лекции	Практические занятия	
	<b>Раздел 1. Дополнительные разделы дискретной математики</b>	<b>80</b>	<b>12</b>	<b>22</b>	<b>46</b>
1.	Однородные рекуррентные соотношения и системы соотношений	14	2	4	8
2.	Неоднородные рекуррентные соотношения и системы соотношений	14	2	4	8
3.	Производящие функции	14	2	4	8
4.	Методы доказательства оценок сложности рекурсивных алгоритмов	10	2	2	6
5.	Простые числа. Асимптотический закон распределения простых чисел	14	2	4	8
6.	Модулярная арифметика. Китайская теорема об остатках	14	2	4	8
	<b>Раздел 2. Графовые и потоковые алгоритмы</b>	<b>62</b>	<b>10</b>	<b>12</b>	<b>40</b>
7.	Алгоритм поиска максимального паросочетания в двудольном графе	10	2	2	8
8.	Алгоритм поиска совершенного паросочетания в двудольном графе	10	2	2	8
9.	Венгерский алгоритм для решения задачи о назначении	10	2	2	8
10.	Алгоритм поиска максимального потока в сети	12	2	4	8
11.	Транспортная задача	10	2	2	8



	<b>Раздел 3. Алгоритмы теории расписаний</b>	<b>60</b>	<b>10</b>	<b>10</b>	<b>40</b>
12.	Алгоритмы теории расписаний для независимых заданий	10	2	2	8
13.	Алгоритмы теории расписаний для зависимых единичных заданий	10	2	2	8
14.	Алгоритмы теории расписаний для зависимых заданий произвольной длительности	10	2	2	8
15.	Алгоритмы теории расписаний для исполнителей с разной производительностью	10	2	2	8
16.	Конвейерная задача	10	2	2	8
	<b>Раздел 4. Алгоритмы комбинаторной оптимизации</b>	<b>52</b>	<b>8</b>	<b>12</b>	<b>32</b>
17.	Динамическое программирование	12	2	4	8
18.	Дискретная задача оптимального распределения инвестиций	8	2	4	8
19.	Метод ветвей и границ	10	2	2	8
20.	Сведение задач комбинаторной оптимизации к задачам на потоки	8	2	2	8
	<b>Раздел 5. Приближенные алгоритмы и эвристики</b>	<b>50</b>	<b>6</b>	<b>10</b>	<b>34</b>
21.	Приближенные алгоритмы и эвристики для решения задачи о рюкзаке	8	2	2	8
22.	Приближенные алгоритмы и эвристики для решения задачи коммивояжера	8	2	2	8
23.	Генетические алгоритмы	28	2	6	18
	<i>Всего:</i>	<b>304</b>	<b>46</b>	<b>66</b>	<b>192</b>

## 6 Формы контроля знаний студентов

Тип контроля	Форма контроля	2 год				Параметры
		1	2	3	4	
Текущий (неделя)	Контрольная работа		*			Письменная работа 80 мин.
Итоговый	Экзамен				*	Письменный экзамен 80 мин.

## 7 Критерии оценки знаний, навыков

При выполнении контрольной работы студент должен продемонстрировать знания основных понятий и алгоритмов из соответствующего раздела учебного курса, умения применять указанные алгоритмы для решения предложенных задач и обосновывать корректность полученных решений. Количество задач в контрольных работах – от 6 до 8. Каждая задача оценивается в 1-2 балла, так что общая сумма баллов равна 10.

При выполнении индивидуальных практических заданий студент должен продемонстрировать знания основных понятий и алгоритмов из соответствующего раздела учебного курса, умения самостоятельно изучать учебную литературу и применять полученные знания при решении предложенных задач.

Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

При выполнении письменной экзаменационной работы студент должен продемонстрировать знания основных понятий и алгоритмов из всего учебного курса, умения применять указанные алгоритмы для решения предложенных задач и обосновывать корректность полученных решений. Работа содержит 1 теоретический вопрос, который оценивается в 2 балла, и 5 практических заданий, каждое из которых оценивается в 1-2 балла, так что общая сумма баллов равна 10.



Промежуточный и итоговый контроль включает как теоретические вопросы, так и практические задания по темам всего курса. Примерный перечень вопросов для подготовки к экзамену по дисциплине приведен в разделе 10.1.

## 8. Содержание дисциплины

### 1. Раздел 1. Дополнительные разделы дискретной математики

Тема 1. Однородные рекуррентные соотношения и системы

Решение однородных систем линейных рекуррентных соотношений с постоянными коэффициентами методом исключения неизвестных и матричным методом (через нахождение собственных значений и собственных векторов матрицы, составленной из коэффициентов).

Тема 2. Неоднородные рекуррентные соотношения и системы

Решение неоднородных систем линейных рекуррентных соотношений с постоянными коэффициентами методом исключения неизвестных и матричным методом (через нахождение собственных значений и собственных векторов матрицы, составленной из коэффициентов). Примеры задач, приводящих к рекуррентным соотношениям.

Тема 3. Производящие функции

Полиномиальные и экспоненциальные производящие функции. Нахождение производящих функций для конкретных последовательностей с использованием рядов Тейлора и известных разложений элементарных функций в ряд Маклорена. Решение линейных рекуррентных соотношений с постоянными коэффициентами через производящие функции.

Тема 4. Методы доказательства оценок сложности рекурсивных алгоритмов

Примеры рекурсивных алгоритмов. Получение рекуррентных соотношений при анализе сложности рекурсивных алгоритмов. Взаимная рекурсия. Решение линейных неоднородных рекуррентных соотношений с постоянными коэффициентами (методом неопределенных коэффициентов) и с переменными коэффициентами (через производящие функции).

Тема 5. Простые числа. Асимптотический закон распределения простых чисел

Элементарные методы проверки простоты чисел: метод проб, решето Эратосфена, метод Ферма. Числа Мерсенна и Ферма. Теорема Чебышева о распределении простых чисел.

Тема 6. Модулярная арифметика. Китайская теорема об остатках

Алгоритм Евклида. НОД и НОК. Сравнения. Классы вычетов. Обратный мультипликативный элемент. Функция Эйлера. Малая теорема Ферма. Теорема Эйлера. Алгоритмы модулярной арифметики. Системы сравнений. Китайская теорема об остатках.

### 2. Раздел 2. Графовые и потоковые алгоритмы

Тема 7. Алгоритм поиска максимального паросочетания в двудольном графе

Чередующиеся цепи. Метод «волны» для поиска чередующихся цепей. Теорема Берга. Алгоритм поиска максимального паросочетания с помощью чередующихся цепей.

Тема 8. Алгоритм поиска совершенного паросочетания в двудольном графе

Чередующееся дерево. Метод «волны» для поиска чередующихся деревьев. Теорема Холла. Алгоритм поиска совершенного паросочетания с помощью чередующихся деревьев.

Тема 9. Венгерский алгоритм для решения задачи о назначении



Матрица затрат и её свойства. Диагональная редукция матрицы затрат. Венгерский алгоритм для решения задачи о назначении.

Тема 10. Алгоритм поиска максимального потока в сети

Поток в двухполюсной сети. Величина потока. Разрез сети. Пропускная способность разреза. Теорема Форда-Фалкерсона. Остаточная сеть. Алгоритм поиска максимального потока в двухполюсной сети с помощью остаточной сети. Стоимость потока в двухполюсной сети. Остаточная сеть. Алгоритм поиска максимального потока минимальной стоимости путем устранения циклов отрицательной стоимости в остаточной сети.

Тема 11. Транспортная задача

Транспортная задача.

### 3. Раздел 3. Алгоритмы теории расписаний

Тема 12. Алгоритмы теории расписаний для независимых заданий

Построение расписания минимальной стоимости для независимых заданий и одного исполнителя. Построение расписания минимальной длительности с прерываниями для независимых заданий. Ленточная стратегия. Диаграммы Ганта.

Тема 13. Алгоритмы теории расписаний для зависимых единичных заданий.

Построение расписания минимальной длительности без прерываний для заданий с графом зависимости в виде ориентированного корневого дерева. Построение расписания минимальной длительности без прерываний с двумя исполнителями для единичных заданий с произвольным орграфом зависимости. Уровневая стратегия.

Тема 14. Алгоритмы теории расписаний для зависимых заданий произвольной длительности

Построение расписания минимальной длительности с прерываниями для заданий произвольной длительности с произвольным орграфом зависимости. Уровневая стратегия с лексикографическим упорядочением.

Тема 15. Алгоритмы теории расписаний для исполнителей с разной производительностью

Построение расписания минимальной длительности с прерываниями для независимых заданий произвольной длительности и для исполнителей с разной производительностью.

Тема 16. Конвейерная задача

Построение расписания минимальной длительности для независимых двухэтапных заданий произвольной длительности и двух исполнителей (конвейерная задача). Алгоритм Джонсона.

### 4. Раздел 4. Алгоритмы комбинаторной оптимизации

Тема 17. Динамическое программирование

Принцип Бэллмана и метод динамического программирования. Примеры и оценка сложности алгоритмов комбинаторной оптимизации, основанных на методе динамического программирования.

Тема 18. Дискретная задача оптимального распределения инвестиций

Полиномиальный алгоритм оптимального дискретного распределения инвестиций на основе метода динамического программирования. Комбинаторная оценка его сложности.

Тема 19. Метод ветвей и границ





Метод ветвей и границ. Алгоритмы для решения задачи о рюкзаке и задачи коммивояжера на основе метода ветвей и границ. Алгоритм Литтла.

Тема 20. Сведение задач комбинаторной оптимизации к задачам на потоки

Возможность сведения некоторых задач комбинаторной оптимизации к задачам на потоки. Решение задач о кратчайших путях в графе, о максимальном паросочетании и задачи о назначении с помощью потоковых алгоритмов.

## 5. Раздел 5. Приближенные алгоритмы и эвристики

Тема 21. Приближенные алгоритмы и эвристики для решения задачи о рюкзаке

Эвристики и приближенный алгоритм для задачи о рюкзаке с гарантированной погрешностью. Точный полиномиальный алгоритм для задачи о сверхвозрастающем рюкзаке.

Тема 22. Точные и приближенные алгоритмы для решения задачи коммивояжера

Эвристики для решения задачи коммивояжера. Приближенный алгоритм Кристофидиса для евклидовой задачи коммивояжера с гарантированной точностью.

Тема 23. Генетические алгоритмы

Общая схема и параметры генетического алгоритма. Операторы скрещивания, мутации и отбора. Решение задачи коммивояжера, задачи о рюкзаке и непрерывной задачи об оптимальном распределении инвестиций с помощью генетического алгоритма. Криптоанализ шифра с помощью генетического алгоритма.

## 9. Образовательные технологии

На практических занятиях студенты работают в мини-группах по 3-4 человека в компьютерном классе. Задача группы, как правило, разработать программный продукт на основе одного из алгоритмов, изучаемых в данном курсе, подготовить набор тестов и выполнить тестирование разработанного программного продукта на этом наборе тестов.

### а. Методические рекомендации преподавателю

На лекциях рекомендуется иллюстрировать материал примерами, показывающими все этапы решения задачи: формализацию и построение математической модели, разработку алгоритма, доказательство его корректности, нахождение его сложности, написание программы, её отладку и тестирование, выполнение программы на реальных входных данных.

На практических занятиях используются следующие методы обучения и контроля усвоения материала: разработка алгоритмов для конкретных задач и их практическая реализация в виде программного продукта с последующей отладкой и тестированием.

При оценке выполненных заданий особое внимание рекомендуется обратить на оценки разработанных программ (получение теоретических оценок, сравнение проведенных экспериментов с результатами, полученными различными способами).

### б. Методические указания студентам

Студенту рекомендуется следующая схема выполнения домашних заданий:

- проработать конспект лекций;
- проанализировать основную и дополнительную литературу, рекомендованную по изучаемому разделу;
- проанализировать варианты решений, предложенные преподавателем на практических занятиях;
- при затруднениях сформулировать вопросы к преподавателю.





При подготовке рекомендуется использовать электронные учебные материалы, имеющиеся в LMS и медиатеке.

Выполненные задания должны сопровождаться данными анализа результатов (теоретическими оценками, сравнением результатов, полученных различными способами).

## 10. Оценочные средства для текущего контроля и аттестации студента

### а. Тематика заданий текущего контроля

Примерные задания для контрольной работы:

1. Решить систему рекуррентных уравнений

$$\begin{cases} x_{n+1} = 2x_n - 2y_n, \\ y_{n+1} = 2x_n - 3y_n, \\ x_0 = 1, \\ y_0 = 0. \end{cases}$$

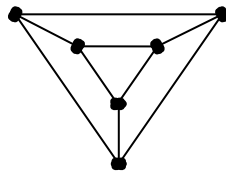
2. Решить рекуррентное уравнение

$$a_{n+2} - 2a_{n+1} + a_n = (4n + 4) \cdot 3^n, \quad a_0 = 1, \quad a_1 = 4.$$

3. С помощью полиномиальной производящей функции решить рекуррентное уравнение

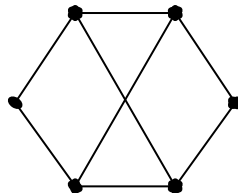
$$(n + 1)a_{n+1} = 2na_n + 3^n, \quad a_0 = 1.$$

4. Используя первую теорему Пойа, найти количество различных раскрасок *граней* плоского графа (включая *внешнюю грань*) с помощью трех красок



- а) с группой автоморфизмов *на плоскости*;  
б) с группой автоморфизмов *в пространстве*.

5. Вершины графа раскрашивают в два цвета так, что 2 вершины оказываются покрашенными в один цвет, а остальные 4 – в другой цвет. Используя вторую теорему Пойа, найти число различных (с точностью до автоморфизма *в пространстве*) раскрасок вершин графа.



### б. Примерный набор индивидуальных заданий:

#### Задание №1

Реализовать в виде программного продукта алгоритм поиска максимального потока минимальной стоимости в двухполюсной сети. Программа должна иметь интуитивно понятный интерфейс, корректно работать для сетей с числом вершин не более 100 вершин. Подготовить набор из 10 тестов разной сложности с заранее известным ответом.



## Задание №2

Реализовать в виде программного продукта алгоритм поиска расписания минимальной длительности с прерываниями для системы независимых заданий и заданного числа идентичных исполнителей, если известны длительности заданий. Программа должна иметь интуитивно понятый интерфейс, выдавать ответ в виде диаграммы Ганта, корректно работать системы с числом заданий не более 1000 и числом исполнителей не более 500. Подготовить набор из 10 тестов разной сложности с заранее известным ответом.

### с. Вопросы для оценки качества освоения дисциплины

1. Однородные системы линейных рекуррентных соотношений с постоянными коэффициентами, их решение методом исключения неизвестных, примеры.
2. Однородные системы линейных рекуррентных соотношений с постоянными коэффициентами, их решение матричным методом, примеры.
3. Неоднородные рекуррентные соотношения, пример нахождения хроматического многочлена для графа-цикла.
4. Полиномиальные производящие функции, примеры полиномиальных производящих функций для простейших последовательностей.
5. Решение рекуррентных соотношения с переменными коэффициентами с помощью полиномиальных производящих функций, примеры.
6. Операции над полиномиальными производящими функциями, свертка.
7. Экспоненциальные производящие функции, вывод верхней оценки для сложности вычисления детерминанта «по определению» и разложением по строке.
8. Симметрическая группа подстановок, теорема Бернсайда, задача об ожерелье на плоскости и в пространстве.
9. Разложение подстановки в произведение циклов, цикловой индекс подстановки и группы подстановок, первая теорема Пойа.
10. Вторая теорема Пойа, задача об окраске граней куба.
11. Задача о числе неэквивалентных химических деревьев и о числе различных булевых функций от  $n$  аргументов.
12. Максимальное паросочетание, теорема Бержа, алгоритм чередующихся цепей, метод «волны» для поиска чередующихся цепей в двудольном графе.
13. Совершенное паросочетание, теорема Холла, метод «волны» для поиска чередующихся деревьев, поиск совершенного паросочетания в двудольном графе.
14. Потоки в сетях, задача о максимальном потоке, разрез сети, теорема Форда–Фолкерсона.
15. Остаточная сеть, её использование для поиска максимального потока в двухполюсной сети.
16. Задача о максимальном потоке минимальной стоимости, остаточная сеть, метод удаления циклов отрицательной стоимости в остаточной сети.
17. Задача о назначениях, венгерский алгоритм.
18. Полиномиальные алгоритмы из теории расписаний: построение расписания минимальной стоимости для независимых заданий и одного исполнителя.
19. Полиномиальные алгоритмы из теории расписаний: построение расписания минимальной длительности для единичных заданий с древовидным графом предшествования.
20. Полиномиальные алгоритмы из теории расписаний: построение расписания минимальной длительности с прерываниями для независимых заданий.
21. Задача о рюкзаке. Метод ветвей и границ для точного решения задачи о рюкзаке (два способа ветвления).
22. Задача о рюкзаке. Приближенный алгоритм для задачи о рюкзаке с гарантированной погрешностью.
23. Задача коммивояжера. Метод ветвей и границ, алгоритм Литтла.
24. Задача коммивояжера. Приближенные алгоритмы. Алгоритм Кристофидиса.



25. Точные и «жадные» алгоритмы и эвристики.
26. Общая схема, операторы и параметры генетического алгоритма. Решение задачи коммивояжера с помощью генетического алгоритма.
27. Оценка эффективности комбинаторного алгоритма, полиномиальные и экспоненциальные алгоритмы, их принципиальные отличия.
28. Нижние оценки сложности комбинаторных алгоритмов. Методы их доказательства. Оптимальные алгоритмы сортировки и поиска.
29. Элементарные методы проверки простоты чисел: метод проб, решето Эратосфена, метод Ферма.
30. Алгоритм Евклида. НОД и НОК, их свойства.
31. Сравнения, их свойства. Классы вычетов. Полная система вычетов. Обратный мультипликативный элемент.
32. Функция Эйлера. Малая теорема Ферма. Теорема Эйлера.
33. Системы линейных сравнений. Китайская теорема об остатках.
34. Криптографическая система RSA. Выбор параметров системы RSA.
35. Возможные атаки на криптосистему RSA.

## 11. Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов на практических занятиях: студенты работают по карточкам, на которых указаны практические задания. Работа каждого студента оценивается с учетом количества задач, решенных им во время занятия. Оценки за работу на практических занятиях преподаватель выставляет в рабочую ведомость. Оценка по 10-ти балльной шкале за работу на практических занятиях определяется перед промежуточным или итоговым контролем и называется –  $O_{\text{контактная}}$ .

**Оценка за текущий контроль** ( $O_{\text{текущий}}$ ) рассчитывается как взвешенная сумма всех форм текущего контроля.

$$O_{\text{текущий}} = 1 * O_{\text{к/р}}$$

**Накопленная оценка** за текущий контроль учитывает результаты студента по текущему контролю следующим образом:

$$O_{\text{накопленная}} = 1/3 * O_{\text{текущий}} + 1/3 * O_{\text{контактная}} + 1/3 * O_{\text{самостоятельная}}$$

Способ округления накопленной оценки текущего контроля: арифметический.

**Результирующая оценка** рассчитывается следующим образом

$$O_{\text{результующая}} = 0,6 * O_{\text{накопленная}} + 0,4 * O_{\text{экз}}$$

Способ округления накопленной оценки промежуточного контроля в форме экзамена: арифметический.

На передаче студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

На экзамене студент может получить дополнительный вопрос (дополнительную практическую задачу, решить к передаче домашнее задание), ответ на который оценивается в 1 балл.

## 12. Учебно-методическое и информационное обеспечение дисциплины

### 12.1. Основная литература

1. Структуры и алгоритмы обработки данных: Учебное пособие / Колдаев В.Д. - М.:ИЦ РИОР, НИЦ ИНФРА-М, 2014. - 296 с.: 60x90 1/16. - (Высшее образование: Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-369-01264-2 - Режим доступа: <http://znanium.com/catalog/product/418290>



## **12.2. Дополнительная литература**

2. Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - М.:КУРС, НИЦ ИНФРА-М, 2016. - 240 с.: 60x90 1/16. - (Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-906818-25-6 - Режим доступа: <http://znanium.com/catalog/product/551224>
3. Математическая логика: Учебное пособие / Игошин В.И. - М.:НИЦ ИНФРА-М, 2016. - 399 с.: 60x90 1/16. - (Высшее образование: Бакалавриат) (Переплёт) ISBN 978-5-16-011691-4 - Режим доступа: <http://znanium.com/catalog/product/539674>

## **12.3.Справочники, словари, энциклопедии**

Справочная система MSDN

## **12.4.Программные средства**

Для успешного освоения дисциплины, студент использует следующие программные средства:

- Среда программирования MS Visual Studio 2015.
- Интегрированный пакет Microsoft Office (MS Word и Excel).
- Средства, обеспечивающие возможность доступа к материалам для подготовки к занятиям в различных форматах (документы MS Word, документы в форматах PDF, HTML, презентации MS Power Point), размещенные на сервере, доступные в Internet.

## **12.5.Дистанционная поддержка дисциплины**

Дистанционная поддержка курса предусмотрена в рамках информационной образовательной среды LMS.

## **13.Материально-техническое обеспечение дисциплины**

Для проведения лекционных занятий используется компьютер с установленным программным обеспечением для демонстрации презентаций и проектор. Практические занятия проводятся в компьютерных классах с установленным программным обеспечением, перечисленным выше.